1. A computer implemented process for managing exceptions throwable during execution of methods in one or more classes by a machine, each method including an exception handler array defining exception handlers associated with the method, the process comprising:

combining the exception handler arrays for two or more methods into a single exception handler table.

2. The process of claim 1 including combining all exception handler arrays for all methods in a class in the single exception handler table.

3. The process of claim 1 including combining all exception handler arrays for all methods in all classes in the single exception handler table.

4. The process of claim 1 including combining all exception handler arrays for all methods in a Java package in the single exception handler table.

5. The process of claim 1 where a method is included in a class file and the step of combining all exception handler arrays includes combining the exception handler arrays for all methods in a class file in the single exception handler table.

6. The process of claim 1 further including

searching the exception handler table when an exception is thrown while executing one of the methods including locating a first matching exception in the single exception handler table.

7. The process of claim 6 where the searching step includes retrieving in order exception handler entries from the exception handler table and checking the type and range of each exception handler for the first matching exception handler.

8. The process of claim 7 further comprising stopping searching if a current exception handler does not match and is the last handler for the top most level of protected code in an associated method.

9. The process of claim 1 where the class files are Java class files.

10. The process of claim 1 where the methods in one or more classes are grouped in a package where the package includes a package data structure including first and second portions, the process including storing the exception handler table in the first portion of the package and all methods in the second portion of the package.

11. The process of claim 10 where the step of combining includes concatenating the exception handler arrays including loading each exception handler array into the first portion of the package data structure in accordance with a predefined ordering.

12. The process of claim 11 where the predefined ordering is determined based on the ordering of methods stored in the second portion of the package data structure.

13. The process of claim 1 where the machine is a virtual machine implemented on a resource-constrained device.

14. The process of claim 13 where the resource-constrained device is a smart card.

15. The process of claim 14 where the methods in one or more classes are grouped in a package and the package is installed on the smart card.

16. The process of claim 15 further including creating a package where the package includes a package data structure including first and second portions, the process including concatenating the exception handler arrays for each of the methods into a exception handler table, storing the exception handler table in the first portion of the package and all methods in the second portion of the package.

17. A method minimizing the amount of storage required for a runtime stack when executing a program, the runtime stack maintained at runtime during the execution of the program by a machine for storing one or more frames where each frame includes a return pointer to an invoking method that called a currently executing method in the program, the method comprising:

combining exception handler information for methods included in the program into a combined exception handler table; and

locating and searching the combined exception handler table when an exception is thrown during execution of one of the methods to locate the exception handler information without requiring the storage on the runtime

stack of a pointer to the exception handler information.

18. The method of claim 17 where the pointer is a direct pointer to the exception handler information.

19. The method of claim 17 where the program is a Java program.

20. The method of claim 19 where the machine is a virtual machine implementing a Java™ virtual machine.

21. The method of claim 17 where the program includes a package of methods, the methods in one or more classes, and where the virtual machine is implemented in a resource constrained device on which the package is installed and executing.

22. The method of claim 21 where the resource constrained device is a smart card.

23. The method of claim 21 further including registering the package in a registry service at installation, the registry service maintaining a pointer and a range, the pointer indicating a location in the resource constrained device of the combined exception handler table associated with a given package, the range defining a range of addresses in the resource constrained device at which methods associated with the package are located.

24. The method of claim 23 where the step of locating includes locating a package associated with a currently executing method including comparing an address at which an exception was thrown against the range for each package

registered in the registry service, the searching step including searching the combined exception handler table associated with a located package.

25. A method of converting class files into a converted applet for execution on a resource constrained device including;

receiving one or more class files, each class file including one or more methods, each method including an exception handler array defining exception handlers catchable by the method;

defining a data structure for storing the methods and exception handlers for the converted applet including a first and second portion;

defining an ordering for the methods and loading the methods according to the ordering in the second portion of the data structure;

combining the exception handler arrays for all methods in a single exception handler table including ordering the exception handler arrays according to the ordering defined for the methods and storing the single exception handler array in the first portion of the data structure.

26. A computer implemented process for managing exceptions throwable during execution of two or more methods in one or more classes by a virtual machine, each method included in a class and including an exception handler array defining exception handlers associated with the method, the individual exception handler arrays combined and forming a single exception handler table for the two or more methods, the process comprising:

searching the exception handler table when an

exception is thrown while executing one of the methods including locating a first matching exception in the single exception handler table.

27. A computer-implemented system for managing exceptions throwable during execution of methods in one or more classes by a machine, each method including an exception handler array defining exception handlers associated with the method, the system comprising instructions to:

combine the exception handler arrays for all methods into a single exception handler table.

28. A computer-implemented system for minimizing the amount of storage required for a runtime stack when executing a program, the runtime stack maintained at runtime during the execution of the program by a machine for storing one or more frames where each frame includes a return pointer to an invoking method that called a currently executing method in the program, the system comprising instructions to:

combine the exception handler information for two or more methods included in the program into a combined exception handler table; and

locate and search the combined exception handler table when an exception is thrown during execution of one of the methods to locate the exception handler information without requiring the storage on the runtime stack of a pointer to the exception handler information.

29. A computer-implemented system for converting class files into a converted applet for execution on a resource constrained device, the system comprising instructions to:

receive one or more class files, each class file including one or more methods, each method including an exception handler array defining exception handlers catchable by the method;

define a data structure for storing the methods and exception handlers for the converted applet including a first and second portion;

define an ordering for the methods and loading the methods according to the ordering in the second portion of the data structure;

combine the exception handler arrays for all methods in a single exception handler table including order the exception handler arrays according to the order defined for the methods and store the single exception handler array in the first portion of the data structure.

30. A computer-implemented system for managing exceptions throwable during execution of methods in one or more classes by a virtual machine, each method in a class described by a class file and including an exception handler array defining exception handlers associated with the method, the individual exception handler arrays combined and forming a single exception handler table for two or more methods, the system comprising instructions to:

search the exception handler table when an exception is thrown while executing one of the two or more methods including locate a first matching exception in the single exception handler table.

ADD A6